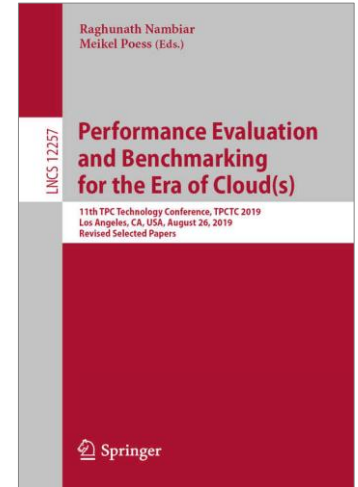


peakmarks® Runs

peakmarks® Version 10.2
February 2024



peakmarks® showcased its software at the 2019 TPC Technology Conference in Los Angeles.



peakmarks® Software and related documentation are protected by intellectual property laws and are subject to a license agreement. Explicit permission is mandatory for any use, modification, distribution, display, transmission, licensing, transfer, publication, or demonstration of the peakmarks® software and its documentation, as stated in the license agreement. Reverse engineering, disassembling, or decompiling of this software is strictly prohibited.

peakmarks® is a registered trademark. Other names may be trademarks of their respective owners.



- 1 Introduction
- 2 Simple and Complex Workloads
- 3 Smart Configuration
- 4 Sample Configuration
- 5 Manual Configuration
- 6 Summary of scripts and commands



Database name	ORA19C / ORA21C / ORA23c
Instance names	ORA19C / ORA21C / ORA23C for a single instance ORA19C1 / ORA21C1 / ORA23C1 for RAC instance 1 ORA19C2 / ORA21C2 / ORA23C2 for RAC instance 2
peakmarks® PDB	PMK
Connect string SYSTEM user	system/manager@SYSAWR
Connect string peakmarks user	bench/bench@PMK
peakmarks® base directory	../pmk



[MBps] megabyte per second

[GBps] gigabyte per second

[dbps] database blocks per second

[rbps] redo blocks per second

[dbpt] database blocks per transaction

[s] seconds

[ms] milliseconds

[μs] microseconds

[IOPS] I/O operations per second

[qps] queries per second

[rps] rows per second

[tps] transactions per second

[kBpt] kilobyte per transaction

[Mops] million operations per second

Nodes number of cluster nodes

Jobs number of workload processes

BuCache Database Buffer Cache

FlCache Database or Exadata Flash Cache

In the following reports, the key performance metrics are marked red.



Simple. Representative. Fast.

Introduction



peakmarks Runs

A series of performance tests processed one after the other is called a peakmarks run

A unique run id identifies each peakmarks run

A peakmarks run is controlled by following commands

```
SQL> exec pmk.start_run (...);
```

```
SQL> exec pmk.stop_run (...);          -> manually stop
```

Each peakmarks run has a status

E – run executing

A – run aborted with pmk.stop_run command

F – run has failures

OK – run successfully processed without failures

Monitoring peakmarks Runs

```
BENCH@PMK SQL> @show_runs
```

```
Thu 03-Aug-2023 20:05:18
```

```
peakmarks Run(s)
```

```
-----
Database....: PMK           Oracle.....: 19.20.0
Instance....: ORA19C1       Build.....: 230801
RAC nodes...: 2            Platfom....: PMEXA01.LAB.LOCAL
```

Run Type	Status	Tests	Run begin	Run end	Elapsed time [min]	Database errors	peakmarks errors	Remark
1	Auto	OK	02-AUG 15:21:24	02-AUG 16:05:33	44.16	0	0	pmk.load_bdb 2 x 2,048 GByte with 2 x 16 loader processes
2	Auto	OK	02-AUG 16:10:29	02-AUG 16:11:14	0.75	0	0	pmk.populate
3	Smart	E	21 02-AUG 16:13:27			0	0	pmk.start_run workload group SRV
4	Auto	OK	02-AUG 16:26:21	02-AUG 16:27:10	0.81	0	0	pmk.populate
5	Smart	E	21 02-AUG 16:27:23			0	0	pmk.start_run workload group SRV
6	Auto	OK	02-AUG 16:43:46	02-AUG 16:44:34	0.79	0	0	pmk.populate
7	Smart	E	21 02-AUG 16:44:53			0	0	pmk.start_run workload group SRV
8	Manual	OK	02-AUG 17:54:27	02-AUG 17:57:46	3.33	0	0	pmk.start_run workload group MANUAL
9	Smart	OK	21 02-AUG 17:58:08	02-AUG 19:07:27	69.33	0	0	pmk.start_run workload group SRV
10	Smart	OK	02-AUG 19:07:28	02-AUG 19:27:19	19.86	0	0	pmk.start_run workload group SRV-MIXED
11	Smart	F	160 02-AUG 19:27:24	02-AUG 22:17:56	170.54	82	492	pmk.start_run workload group STO with increment 4 and parameter 20
12	Smart	F	128 02-AUG 22:18:00	03-AUG 01:58:16	220.26	130	780	pmk.start_run workload group LGWR with increment 8
13	Smart	OK	32 03-AUG 01:58:17	03-AUG 02:14:57	16.66	0	0	pmk.start_run workload group DBWR with increment 4
14	Smart	OK	64 03-AUG 02:14:59	03-AUG 03:20:29	65.50	0	0	pmk.start_run workload group DL with increment 8
15	Smart	OK	74 03-AUG 03:20:31	03-AUG 05:36:06	135.59	0	0	pmk.start_run workload group DA with increment 4
16	Smart	OK	161 03-AUG 05:36:11	03-AUG 12:46:35	430.40	0	0	pmk.start_run workload group TP with increment 4 and parameter 20
17	Smart	OK	32 03-AUG 12:46:37	03-AUG 14:26:29	99.87	0	0	pmk.start_run workload group TP-MIXED1 with increment 4
18	Smart	OK	32 03-AUG 14:26:30	03-AUG 15:27:49	61.32	0	0	pmk.start_run workload group TP-MIXED2 with increment 4
19	Smart	OK	32 03-AUG 15:27:50	03-AUG 17:12:40	104.82	0	0	pmk.start_run workload group PLS
20	Smart	OK	5 03-AUG 17:12:40	03-AUG 17:29:37	16.94	0	0	pmk.start_run workload group PLS-MIXED
21	Manual	OK	14 03-AUG 17:43:11	03-AUG 18:30:28	47.28	0	0	pmk.start_run workload group MANUAL

```
21 rows selected.
```



Performance Tests

Each performance test is described by a set of parameters

- ID unique test number within peakmarks run
- Workload workload of this performance test
- Parameter optional workload parameter
- ALC adaptive load control
if > 0 number of heats to be executed
if 0, execute the test until the runtime target is reached
- Nodes number of RAC nodes used for this performance test
- Jobs number of workload jobs used for this performance test,
distributed over all nodes (parameter Nodes)
- DOP Oracle degree of parallelism for this performance test
- Runtime runtime target in minutes for this specific performance test

Monitoring Performance Test Orders (Queue)

```
BENCH@PMK SQL> @show_orders

Thu 03-Aug-2023 16:59:24

Order Configuration
-----

Database....: PMK                Oracle.....: 19.20.0
Instance....: ORA19C1           Build.....: 230801
RAC nodes...: 2                 Platform....: PMEXA01.LAB.LOCAL

      Runtime
Test Workload  Para  ALC  Nodes  Jobs  DOP  target
              meter                [min]
-----
  1 PLS-BUILTIN  VC    0    2    2    1    3
  2 PLS-BUILTIN  VC    0    2   48    1    3
  3 PLS-BUILTIN  VC    0    2   96    1    3
  4 PLS-BUILTIN  VC    0    2  144    1    3
  5 PLS-BUILTIN  VC    0    2  192    1    3

5 rows selected.

BENCH@PMK SQL>
```



Monitoring Performance Tests

```

BENCH@PMK SQL> @show_tests

Thu 03-Aug-2023 19:19:59

peakmarks Test(s)
-----

Run.....:
Test.....:
Workload....:

Database....: PMK           Oracle.....: 19.20.0
Instance....: ORA19C1       Build.....: 230801
RAC nodes...: 2           Platform....: PMEXA01.LAB.LOCAL


```

Run	Test	Status	Workload	Parameter	ALC	Nodes	Jobs	DOP	Runtime target [min]	Test begin	Test end
9	1	OK	SRV-REPORT	N/A	0	2	2	1	3	02-AUG-2023 18:01:46	02-AUG-2023 18:04:49
	2	OK	SRV-REPORT	N/A	0	2	48	1	3	02-AUG-2023 18:05:01	02-AUG-2023 18:08:07
	3	OK	SRV-REPORT	N/A	0	2	96	1	3	02-AUG-2023 18:08:19	02-AUG-2023 18:11:25
	4	OK	SRV-REPORT	N/A	0	2	144	1	3	02-AUG-2023 18:11:38	02-AUG-2023 18:14:42
	5	OK	SRV-REPORT	N/A	0	2	192	1	3	02-AUG-2023 18:14:56	02-AUG-2023 18:18:01

```

5 rows selected.

BENCH@PMK SQL>

```



peakmarks Runs

There are different types of peakmarks runs dependent on their configuration

- Automatic configuration
- Manual configuration
- Smart configuration
- Sample configuration



Automatic Configuration

The Automatic Configuration is fully automated

It is used for administrative tasks like loading the peakmarks data and populating the database caches



Smart Configuration

The Smart Configuration is fully automated

It is the most convenient and fastest way to get a complete overview of the performance of an Oracle database service

It generates a sequence of max 32 tests for each workload with an increasing number of processes. The increase in the number of processes is defined by the parameter P_INCREMENT when calling the Smart Configuration

peakmarks® automatically terminates the test sequence when this specific workload's most important performance metric stops increasing but runs at least some tests in any case



Sample Configuration

The Sample Configuration is a preconfigured and fully automated peakmarks run

The peakmarks user can customize this configuration

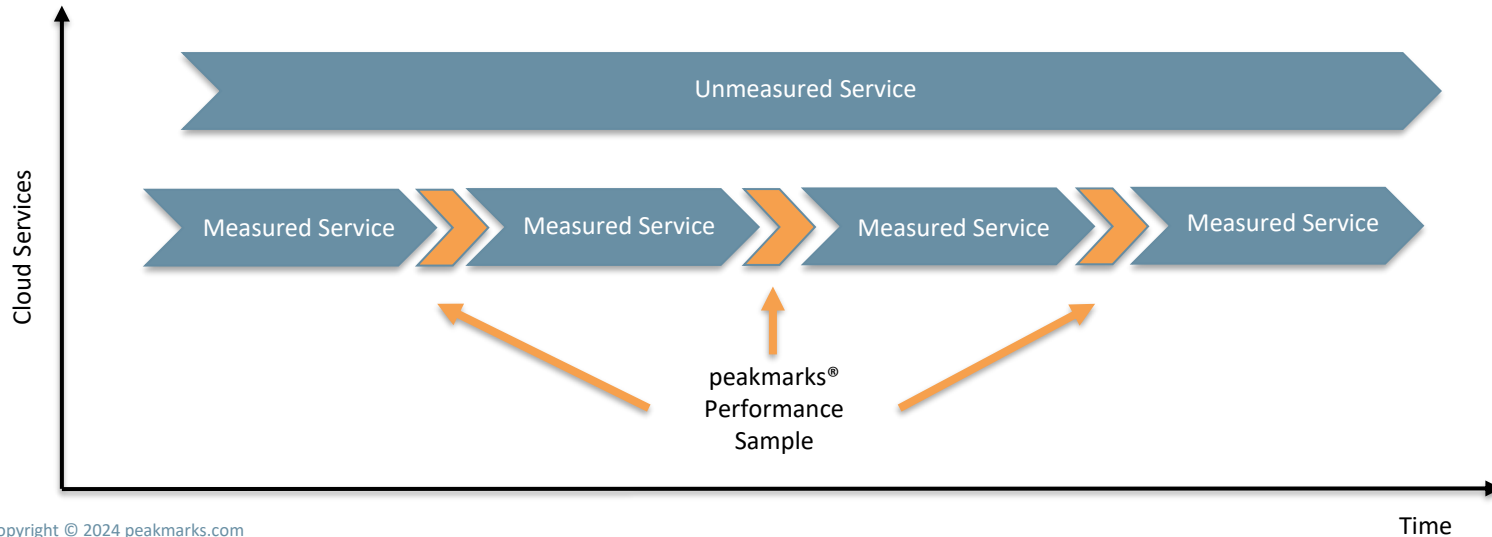
The Sample Configuration is used to take a performance sample of only a few key performance metrics within a short period of time (< 1 hour) during a maintenance window

It is used for quality assurance tasks to ensure that a database service (on-premises or in the cloud) meets the agreed performance metrics over an extended period

Sample Configuration

Repeatedly taking a short performance sample within the maintenance windows (e.g., every month or quarter).

- One peakmarks run takes less than 1 hour
- Determine key performance metrics for selected workloads





Manual Configuration

The manual configuration supports a maximum of configuration options

It is used by engineers to analyze specific load situations or to find configurations for maximum performance numbers

The manual configuration requires the complete input of all test parameters from an external file in CSV format called `manual.csv` located in the directory `../pmk/cfg`



Simple. Representative. Fast.

Simple and Complex Workloads



Simple workloads perform a single task executed by a database job

Complex workloads are composed of several simple workloads executed in parallel by database jobs



Available Workloads

```
BENCH@PMK SQL> @show_workloads
```

```
Mon 31-Jul-2023 08:34:09
```

```
peakmarks Workloads
```

```
-----
```

```
Workload....:
```

```
Type.....:
```

```
Database....: PMK           Oracle.....: 19.20.0
Instance....: ORA19C1       Build.....: 230801
RAC nodes...: 2             Platform...: PMEXA01.LAB.LOCAL
```

Group	Workload	T	Description	Parameter
ADM	ADM-01	S	load peakmarks database (parallel)	N/A
	ADM-02	S	populate buffer caches	N/A

Group	Workload	T	Description	Parameter
DA	DA-COLSTORE	S	perform data analytics query using column store	N/A
	DA-OFFLOAD	S	perform data analytics query using offload technology	N/A
	DA-ROWSTORE	S	perform data analytics query using row store	N/A
	DA-STORAGE	S	perform data analytics query using conventional storage	N/A



Available Workloads (cont.)

Group	Workload	T Description	Parameter
DBWR	DBWR-THR	S test DBWR throughput	N/A
Group	Workload	T Description	Parameter
DL	DL-BUFFER	S perform buffered data load (with different tx sizes)	rpt = {1, 2, 3, 4, 5, 10, 20, 25, 50, 100}
	DL-DIRECT	S perform direct data load	N/A
	DL-STREAM	S perform stream data load (with different tx sizes)	rpt = {1, 2, 3, 4, 5}
Group	Workload	T Description	Parameter
LGWR	LGWR-LAT1	S test LGWR latency with small-sized transactions	N/A
	LGWR-LAT125	S test LGWR latency with large-sized transactions	N/A
	LGWR-LAT25	S test LGWR latency with medium-sized transactions	N/A
	LGWR-THR	S test LGWR throughput	N/A
Group	Workload	T Description	Parameter
PLS	PLS-ADD	S perform PL/SQL simple arithmetic operations	data types = {SI, SF, PI, NU, DA}
	PLS-BUILTIN	S perform PL/SQL built-in operations	data types = {SI, SF, PI, NU, VC}
	PLS-FIBO	S calculate fibonacci number of n	n = {39, 40, 41, 42, 43, 44}
	PLS-MIXED	C perform PL/SQL built-in operations on different data types	N/A
	PLS-PRIME	S calculate first n prime numbers	n = {1000, 1001, ..., 10000}



Available Workloads (cont.)

Group	Workload	T	Description	Parameter
SRV	SRV-MIXED	C	perform mix of lookup queries and scans on row store	N/A
	SRV-QUERY1	S	perform simple query (1 row per query) on row store	N/A
	SRV-QUERY25	S	perform more complex query (avg 25 rows per query) on row store	N/A
	SRV-REPORT	S	perform report (avg 125 rows per report) on row store	N/A
	SRV-SCAN	S	perform full table scan on row store	N/A
	SRV-WARMUP	S	warm-up buffer cache for SRV workloads	N/A
Group	Workload	T	Description	Parameter
STO	STO-OFFLOAD	S	perform sequential read using offload technology	N/A
	STO-PRECON	S	pre-conditioning of flash storage system	N/A
	STO-RANDOM	S	perform random single block read/write	update ratio = {0, 10, ..., 100}
	STO-READ	S	perform sequential read using conventional storage system	N/A
	STO-SCATTER	S	perform scattered write	N/A



Available Workloads (cont.)

Group	Workload	T	Description	Parameter
TP	TP-CUSTOM	C	perform customer-defined transaction mix1	N/A
	TP-HEAVY	S	perform heavy-weighted transactions	update ratio = {0, 10, ..., 100}
	TP-LIGHT	S	perform light-weighted transactions	update ratio = {0, 10, ..., 100}
	TP-LOOKUP	S	perform lookup query (1 row per query)	N/A
	TP-MEDIUM	S	perform medium-weighted transactions	update ratio = {0, 10, ..., 100}
	TP-MIXED1	C	perform mix of read-intensive OLTP transactions	N/A
	TP-MIXED2	C	perform mix of write-intensive OLTP transactions	N/A
	TP-REPORT	S	perform report transactions	N/A
	TP-WARMUP	S	warm-up buffer cache for TP workloads	N/A

39 rows selected.

BENCH@PMK SQL>



Composition of complex workloads

```
BENCH@PMK SQL> @show_complex

Mon 31-Jul-2023 08:46:10

Complex peakmarks Workloads
-----

C Workload...

Database....: PMK           Oracle.....: 19.20.0
Instance....: ORA19C1       Build.....: 230801
RAC nodes...: 2             Platform...: PMEXA01.LAB.LOCAL

Complex      Simple      Para
workload     workload     meter  Jobs
-----
PLS-MIXED    PLS-ADD      PI     1
              PLS-ADD      NU     1
              PLS-BUILTIN  VC     1
              PLS-BUILTIN  NU     1
```



Composition of complex workloads (cont.)

Complex workload	Simple workload	Para meter	Jobs
SRV-MIXED	SRV-QUERY1	N/A	1
	SRV-QUERY25	N/A	1
	SRV-REPORT	N/A	1
	SRV-SCAN	N/A	1
Complex workload	Simple workload	Para meter	Jobs
TP-MIXED1	DL-BUFFER	3	1
	TP-LOOKUP	N/A	1
	TP-MEDIUM	40	1
	TP-REPORT	N/A	1
Complex workload	Simple workload	Para meter	Jobs
TP-MIXED2	DL-BUFFER	2	1
	TP-HEAVY	10	1
	TP-LIGHT	30	1
	TP-MEDIUM	20	1

16 rows selected.

BENCH@PMK SQL>



Simple. Representative. Fast.

Smart Configuration



To run the Smart Configuration, use the procedure `pmk.start_run` with the following parameters

- `P_WKLGROUP` workload group
- `P_INCREMENT` optional step size for some workload groups when increasing processes; default value based on peakmarks configuration parameter `CPUCOUNT`
- `P_PARAMETER` optional workload parameter; default value based on `WKLGROUP`
- `P_REMARK` optional comment



Examples

- SQL> exec pmk.start_run ('TP');
- SQL> exec pmk.start_run ('TP', 2);
- SQL> exec pmk.start_run ('TP', 2, 20)
- SQL> exec pmk.start_run ('TP', 2, 20, 'TP workloads on test system');



Determination of the JOBS sequence

All job sequences start with 1 (single thread performance).

For all CPU-bound workloads, peakmarks® generates 4 additional measuring points:

- Utilize 25% of all threads $\text{JOBS} = 0.25 * \text{CPU_COUNT}$
- Utilize 50% of all threads $\text{JOBS} = 0.50 * \text{CPU_COUNT}$
- Utilize 75% of all threads $\text{JOBS} = 0.75 * \text{CPU_COUNT}$
- Utilize all threads $\text{JOBS} = \text{CPU_COUNT}$

For all other workloads, peakmarks® generates a maximum of 31 additional measuring points based on parameter P_INCREMENT.

The maximum number of JOBS is limited by $3 * \text{CPU_COUNT}$.

Auto Stop of Test Series

peakmarks® stops the configured series of performance tests if the performance does no longer increase

```

BENCH@PMK SQL> @show_teststats

Thu 03-Aug-2023 19:20:13

peakmarks Test Statistics
-----

Run.....: 11
Test.....:
Workload....:

Database....: PMK           Oracle.....: 19.20.0
Instance....: ORA19C1       Build.....: 230801
RAC nodes...: 2           Platform....: PMEXA01.LAB.LOCAL

Run Test Status Workload  Para  KPM value  KPM value  KPM value  PMKSNP1#  PMKSNP2#  DBSSNP1#  DBSSNP2#
meter  curr test  prev test  inc [%]
-----
  11   1   OK STO-OFFLOAD    0  47,113,456,751    0    0.00    80    81    1486    1487
      2   OK STO-OFFLOAD    0  67,427,063,725  47,113,456,751  21.56    82    83    1488    1489
      3   OK STO-OFFLOAD    0  72,479,649,898  67,427,063,725   3.75    84    85    1490    1491
      4   OK STO-OFFLOAD    0  75,910,606,534  72,479,649,898   2.37    86    87    1492    1493
      5   OK STO-OFFLOAD    0  75,437,492,237  75,910,606,534  -0.31    88    89    1494    1495

5 rows selected.

BENCH@PMK SQL>
    
```



Workload groups

- SQL> exec pmk.start_run ('SRV');
- SQL> exec pmk.start_run ('STO', 4, 20);
- SQL> exec pmk.start_run ('LGWR', 4, NULL, 'My comment');
- SQL> exec pmk.start_run ('DBWR');
- SQL> exec pmk.start_run ('DL', 4, Null, 'My comment');
- SQL> exec pmk.start_run ('DA', 4);
- SQL> exec pmk.start_run ('TP', 4, 20);
- SQL> exec pmk.start_run ('PLS');

- SQL> exec pmk.start_run ('ALL');

Complex workloads . . .

- SQL> exec pmk.start_run ('SRV-MIXED');
- SQL> exec pmk.start_run ('TP-MIXED1');
- SQL> exec pmk.start_run ('TP-MIXED2');
- SQL> exec pmk.start_run ('PLS-MIXED');



The workload group **ALL** consists of the following workloads:

- Complex workload SRV-MIXED
- Workload group STO
- Workload group LGWR
- Workload group DBWR
- Workload group DL
- Workload group DA
- Complex workload TP-MIXED1
- Complex workload TP-MIXED2
- Complex workload PLS-MIXED

Start the workload group ALL with the following command

- `SQL> exec pmk.start_run ('ALL', 4, 20, 'First test series');`



Some comments

Server (SRV) workloads

- Parameters P_PARAMETER and P_INCREMENT ignored

Storage (STO) workloads

- Parameter P_PARAMETER only used for an optional second set of STO_RANDOM performance tests
- Workload STO-OFFLOAD on Exadata only

Log Writer (LGWR) workloads

- Parameter P_PARAMETER ignored



Some comments

Database Writer (DBWR) workload

- Parameter P_PARAMETER ignored

Data Load (DL) workloads

- Parameter P_PARAMETER is only available for workloads DL-BUFFER and DL-STREAM and specifies the number of rows per insert transaction
- Workload DL-STREAM is only available when the memory-optimized pool is configured

Data Analytics (DA) workloads

- Parameter P_PARAMETER ignored
- Parameter P_INCREMENT ignored for workloads DA-ROWSTORE and DA-COLSTORE
- Workload DA-COLSTORE only when column-store is populated



Some comments

Transaction processing (TP) workloads

- Parameter P_PARAMETER uses default 20 if no value is specified
- If the memory-optimized pool is configured, the workload TP-LOOKUP accesses rows via hash key, otherwise the workload TP-LOOKUP accesses rows via b-tree index

PL/SQL Application (PLS) workloads

- Parameter P_PARAMETER and P_INCREMENT ignored



Simple. Representative. Fast.

Sample Configuration



The sample is a pre-configured and fully automated peakmarks run

The configuration is based on CPU_COUNT and stored in table PMK_SAMPLE

The peakmarks user can customize the configuration and change the contents of table PMK_SAMPLE using SQL commands



Pre-configured Sample Configuration

```

BENCH@PMK SQL> @show_sample

Mon 31-Jul-2023 09:11:49

Sample peakmarks Configuration
-----

Database....: PMK           Oracle.....: 19.20.0
Instance....: ORA19C1       Build.....: 230801
RAC nodes...: 2            Platform...: PMEXA01.LAB.LOCAL

Test Workload      Para
meter             ALC Nodes  Jobs  DOP  Runtime
target           Remark
[min]
-----
 1 SRV-MIXED       N/A    0    1    1    1    3 mixed queries and scans on cached data
 2 STO-READ        N/A    0    1    3    4    3 SQL sequential read
 3 STO-OFFLOAD     N/A    0    1    3    1    3 SQL sequential read offload
 4 STO-RANDOM      0      0    1    3    1    3 SQL random read 100% read
 5 STO-RANDOM      20     0    1    3    1    3 SQL random read 80% read 20% write
 6 LGWR-LAT1       N/A    0    1    3    1    3 log writer latency
 7 LGWR-THR        N/A    0    1    3    1    3 log writer throughput
 8 DL-BUFFER       5      0    1    3    1    3 buffered data load
 9 DL-DIRECT       125000 0    1    3    1    3 direct data load
10 TP-MIXED1       N/A    0    1    1    1    5 mix of read-intensive OLTP transactions and reports
11 TP-MIXED2       N/A    0    1    1    1    5 mix of write-intensive OLTP transactions and reports
12 PLS-MIXED      N/A    0    1    1    1    3 mixed PL/SQL operations on mixed datatypes

Sum                                     40

12 rows selected.

BENCH@PMK SQL>
    
```



Examples

- SQL> exec pmk.start_run ('sample');



Swiss precision in timing.

Manual Configuration



Implemented as external table manual.csv

- Location ../pmk/cfg
- Each row describes one performance test

The ../pmk/cfg directory contains some templates for

- Adapt the template and copy it to manual.csv

Check the current configuration file manual.csv with

- SQL> @show_manual



Configuration file manual.csv is managed as an external table

- Column A Workload workload name
- Column B Parameter workload parameter
- Column C ALC adaptive load control, if 0, use peakmarks® configuration parameter RUNTIME otherwise the number of heats
- Column D Nodes number of RAC nodes used for this test
- Column E Jobs number of jobs used for this test, distributed over all nodes configured in column D
- Column F DOP Oracle degree of parallelism for this test
- Column G Runtime overrides configuration parameter RUNTIME for this specific test
- Column H Comment additional information



Monitoring the manual.csv configuration file

```
BENCH@PMK SQL> @show_manual
```

```
Mon 31-Jul-2023 09:14:11
```

```
Manual peakmarks Configuration
```

```
-----
```

```
Database....: PMK                Oracle.....: 19.20.0  
Instance....: ORA19C1            Build.....: 230801  
RAC nodes...: 2                  Platform...: PMEXA01.LAB.LOCAL
```

Test Workload	Para meter	ALC	Nodes	Jobs	DOP	Runtime target [min]	Remark
1 PLS-ADD	SI	0	1	1	1	1	datatype SIMPLE_INTEGER

```
BENCH@PMK SQL>
```




Some comments on Column F DOP

When DOP is set to 0 or 1

- No parallel query enabled
- No direct I/O enabled

When DOP is set > 1

- Parallel query and direct I/O enabled, even if DOP = 1 for following workloads
 - » STO-READ
 - » DA-STORAGE, DA-OFFLOAD
- Parallel query enabled
 - » SRV-SCAN (no direct I/O because data is in row cache)



Manual peakmarks configurations support all kind of parallelism

```
BENCH@PMK SQL> @show_manual
```

Mon 28-Dec-2020 17:23:44

Manual peakmarks Configuration

Test	Workload	Parameter	ALC	Nodes	Jobs	DOP	Runtime target [min]	Remark
1	STO-READ	N/A	0	1	1	1	3	sequential read
2	STO-READ	N/A	0	1	1	2	3	sequential read
3	STO-READ	N/A	0	1	1	4	3	sequential read
4	STO-READ	N/A	0	1	1	1	3	sequential read
5	STO-READ	N/A	0	1	2	1	3	sequential read
6	STO-READ	N/A	0	1	4	1	3	sequential read
7	STO-READ	N/A	0	2	4	1	3	sequential read

-> Intra-SQL Parallelism

-> Inter-SQL Parallelism

-> Cluster Parallelism



Templates

The `../pmk/cfg` directory contains some templates for all workload groups

Just adapt (according to your platform) one of these files copy it to `manual.csv`



Examples

- SQL> exec pmk.start_run ('manual');



Simple. Representative. Fast.

Summary of Scripts and Commands



Manual peakmarks configurations

```
SQL> @show_manual
```

```
SQL> exec pmk.start_run ('manual')
```

Sample peakmarks configurations

```
SQL> @show_sample
```

```
SQL> exec pmk.start_run ('sample')
```

Smart peakmarks configurations

```
SQL> exec pmk.start_run (p_wkgroup,  
                        p_increment, p_parameter, p_comment)
```

Monitoring queue of peakmarks tests

```
SQL> @show_orders
```



peakmarks Mission

Identify Key Performance Metrics for Oracle Database Platforms.

On-Premises and in the Cloud.

For Quality Assurance, Evaluations, and Capacity Planning.